# CLAIM AMENDMENTS

## Claim Amendment Summary

### Claims pending

- Before this Amendment: Claims 1-5, 8-10, 12, 13, 15-26, and 28-38.
- After this Amendment: Claims 1-5, 8, 10, 12-13, 16, 18-26, 29-31, 33-34, and 36-38.

**Non-Elected, Canceled, or Withdrawn claims**: Claims 6-7, 9, 11, 14-15, 17, 27-28, 32, and 35.

**Amended claims**: Claims 1, 12, 20, and 29.

**New claims**: None.

---

## Claims:

**1. (Currently Amended)** One or more processor-accessible storage media comprising processor-executable instructions that, when executed, direct a device to perform actions comprising:

determining if an instruction of a line of common intermediate language code meets a predetermined exception-related criterion; and

if so, injecting a decision point in association with the instruction of the line of common intermediate language code, wherein:

the decision point ~~enabling~~ enables a decision as to whether an exception is to be thrown with respect to the instruction;

~~wherein~~ the action of injecting a decision point comprises an action of injecting a bookmark entry, wherein the action of injecting a decision point further comprises an action of:

injecting a call to a decision runtime library, the decision runtime library comprising a program that is adapted to evaluate whether the exception is to be thrown with respect to the instruction by utilizing at least one throw exception decision (TED) logic factor, wherein the TED logic factor comprises one of the following:

random;

first path, wherein the first path TED logic factor throws an exception when encountering a particular decision point along a new execution path using an identifier of the particular decision point and one or more stack values;

the bookmark entry including an instruction type indicator that indicates an instruction type for the instruction of the line of common intermediate language code.

**2. (Previously Presented)** The one or more processor-accessible storage media as recited in claim 1, comprising the processor-executable instructions that, when executed, direct the device to perform further actions comprising:

retrieving the line of common intermediate language code from a common intermediate language code program prior to the determining;

retrieving another line of common intermediate language code from the common intermediate language code program; and

repeating the action of determining and the action of injecting a decision point for an instruction of the retrieved other line of common intermediate language code.

**3.** **(Previously Presented)** The one or more processor-accessible storage media as recited in claim 1, wherein the action of determining comprises an action of:

determining if the instruction of the line of common intermediate language code is capable of throwing an exception.

**4.** **(Previously Presented)** The one or more processor-accessible storage media as recited in claim 1, wherein the action of determining comprises an action of:

determining if the instruction of the line of common intermediate language code is capable of throwing an exception and is related to a pre-selected exception area.

**5.** **(Previously Presented)** The one or more processor-accessible storage media as recited in claim 1, wherein the action of determining comprises an action of:

determining if the instruction of the line of common intermediate language code is capable of throwing an exception with reference to a common intermediate language code specification.

**6. (Canceled)**

**7. (Canceled)**

**8. (Previously Presented)** The one or more processor-accessible storage media as recited in claim 1, wherein the bookmark entry

further includes an identifier that uniquely identifies the decision point within the common intermediate language code that is being instrumented.

**9. (Canceled)**

**10. (Previously Presented)** The one or more processor-accessible storage media as recited in claim 1, wherein at least a portion of the processor-executable instructions comprise at least part of an instrumentation tool that produces instrumented common intermediate language code from the common intermediate language code by repeating the actions of determining and injecting a decision point for a plurality of respective instructions of a plurality of respective lines of the common intermediate language code.

**11. (Canceled)**

**12. (Currently Amended)** A device comprising:

a processor;

a memory device, the memory device encoded with:

instrumented common intermediate language code that includes a test couplet corresponding to a decision point and an associated instruction, the associated instruction capable of causing a fault;

a decision runtime library that is adapted to evaluate the test couplet to selectively decide whether to throw an exception ~~with respect to the associated instruction~~ responsive to a bookmark entry and based on at least one throw exception decision (TED) logic factor selected from the group comprising:

throwing an exception randomly; and

throwing an exception when encountering a particular decision point along each new execution path using an identifier of the particular decision point and one or more stack values; and

a common language runtime component that interprets the decision point so as to call the decision runtime library prior to executing the associated instruction;

wherein the decision point comprises [[a]] the bookmark entry and a call to the decision runtime library, and wherein the bookmark entry comprises an indication of an instruction type of the associated instruction and an identifier of the decision point.

**13. (Original)** The device as recited in claim 12, wherein the instrumented common intermediate language code is in a binary form.

**14. (Canceled)**

**15. (Canceled)**

**16. (Previously Presented)** The device as recited in claim 12, wherein the decision runtime library is further adapted to evaluate the test couplet to selectively decide whether to throw an exception responsive to the bookmark entry and based on throw exception decision logic.

**17. (Canceled)**

**18. (Previously Presented)** The device as recited in claim 12, wherein the decision runtime library is further adapted to evaluate the test couplet to selectively decide whether to throw an exception responsive to the indication of the instruction type of the associated instruction.

**19. (Previously Presented)** The device as recited in claim 12, wherein the decision runtime library is (i) modularized by exception category and/or (ii) operative in dependence on an instruction type of the associated instruction as determinable by the indication of the instruction type from the bookmark entry.

**20. (Currently Amended)** An arrangement for enabling reliability testing of managed code, the arrangement including one or more processor-accessible storage media device; wherein the arrangement comprises:

instrumentation means for instrumenting common intermediate language code with a plurality of decision points to produce instrumented common intermediate language code;

wherein the instrumentation means comprises:

analysis means for analyzing whether individual instructions of a plurality of instructions of the common intermediate language code can result in a failure; and

injection means for injecting a respective decision point in association with each respective individual instruction, which can result in a failure as analyzed by the analysis means, of the plurality of instructions of the common intermediate language code;

wherein the injection means comprises:

means for injecting a call to a decision runtime library, the decision runtime library comprising a program that is adapted to evaluate whether an exception is to be thrown with respect to the instruction by utilizing at least one throw exception decision (TED) logic factor, wherein the TED logic factor comprises one of the following:

random;

first path, wherein the first path TED logic factor throws an exception when encountering a particular decision point along a new execution path using an identifier of the particular decision point and one or more stack values;

means for injecting a respective bookmark entry that indicates an instruction type of the respective individual instruction associated with the respective decision point; and

decision means for deciding whether to throw an exception at each decision point of the plurality of decision points.

**21. (Previously Presented)** The arrangement as recited in claim 20, wherein the respective bookmark entry further identifies the respective decision point.

**22. (Previously Presented)** The arrangement as recited in claim 20, wherein the injection means further comprises:

means for injecting a call at least one module that is capable of evaluating the respective decision point with regard to whether a failure is to be induced.

**23. (Original)** The arrangement as recited in claim 20, further comprising:

common language runtime means for executing the instrumented common intermediate language code and the decision means in a runtime environment.

**24. (Original)** The arrangement as recited in claim 23, wherein the decision means operates while the instrumented common intermediate language code is being executed when the common language runtime means calls the decision means at each decision point of the plurality of decision points.

**25. (Previously Presented)** The arrangement as recited in claim 20, wherein the decision means comprises:

evaluation means for evaluating whether to throw an exception responsive to the respective bookmark entry of each respective decision point of the plurality of decision points and based on at least one throw exception decision logic factor.

**26. (Previously Presented)** The arrangement as recited in claim 20, wherein the arrangement comprises at least one device having the one or more processor-accessible storage media.

**27. (Canceled)**

**28. (Canceled)**

**29. (Currently Amended)** A method for instrumentation injection with regard to a common language runtime environment, the method comprising:

determining whether an instruction from common intermediate language code is capable of causing an exception; and if so, injecting a decision point in association with the instruction to mark the instruction for evaluation during a common language runtime execution, the evaluation involving a decision as to whether a failure is to be induced with respect to the instruction; wherein the injecting a decision point comprises ~~injecting an indicator of an instruction type of the instruction~~ of injecting a bookmark entry, wherein the action of injecting a decision point further comprises an action of:

injecting a call to a decision runtime library, the decision runtime library comprising a program that is adapted to evaluate whether an exception is to be thrown with respect to the instruction by utilizing at least one throw exception decision (TED) logic factor, wherein the TED logic factor comprises one of the following:

random;

first path, wherein the first path TED logic factor throws an exception when encountering a particular decision point along a new execution path using an identifier of the particular decision point and one or more stack values;

the bookmark entry including an instruction type indicator that indicates an instruction type for the instruction of the line of common intermediate language code.

**30. (Original)** The method as recited in claim 29, wherein the determining comprises:

determining whether the instruction from the common intermediate language code is capable of causing an exception and is (i) related to a pre-selected exception category and/or (ii) of a pre-selected instruction type.

**31. (Previously Presented)** The method as recited in claim 29, wherein the injecting a decision point further comprises:

injecting an identifier of the decision point.

**32. (Canceled)**

**33. (Previously Presented)** One or more processor-accessible storage media comprising processor-executable instructions that, when executed, direct a device to perform the method as recited in claim 29.

**34. (Previously Presented)** The method as recited in claim 29, further comprising:

repeating the determining and the injecting a decision point for a plurality of instructions from the common intermediate language code; and

producing instrumented common intermediate language code as a result of the repeating.

**35. (Canceled)**

**36. (Original)** The method as recited in claim 29, further comprising: selectively deciding whether the execution is to fail at the decision point.

**37. (Original)** The method as recited in claim 36, further comprising:

if it is decided at the selectively deciding that the execution is to fail at the decision point, then choosing which exception of at least two exceptions is to be thrown.

**38. (Original)** The method as recited in claim 36, further comprising:

if it is decided at the selectively deciding that the execution is to fail at the decision point, then inducing a failure in the execution of the common language runtime with respect to the instruction.